ECE421: Introduction to Machine Learning — Fall 2024

Assignment 2: Gradient Descent, Multiclass Logistic Regression, and K-Means

Due Date: Friday, October 18, 11:59 PM

General Notes

1. Programming assignments can be done in groups of up to 2 students. Students can be in different sections.
2. Only one submission from a group member is required.
3. Group members will receive the same grade.
4. Please post assignment-related questions on Piazza.

Turning It In

You need to submit your version of the following files:

- `myTorch.py`
- `PA2_qa.pdf` that answer questions related to the implementations.
- The cover file with your name and student ID filled (it can be as the first page of your `PA2_qa.pdf` or as a separate PDF file.)

Please pack them into a single folder, compress into a zip file and name it as PA2.zip. Please submit the zip file to Quercus.

Group Members

Name (and Name on Quercus)	UTORid
Shulin Ji	jishuli1
Ruoheng Wang	wangr279

1 Gradient Descent

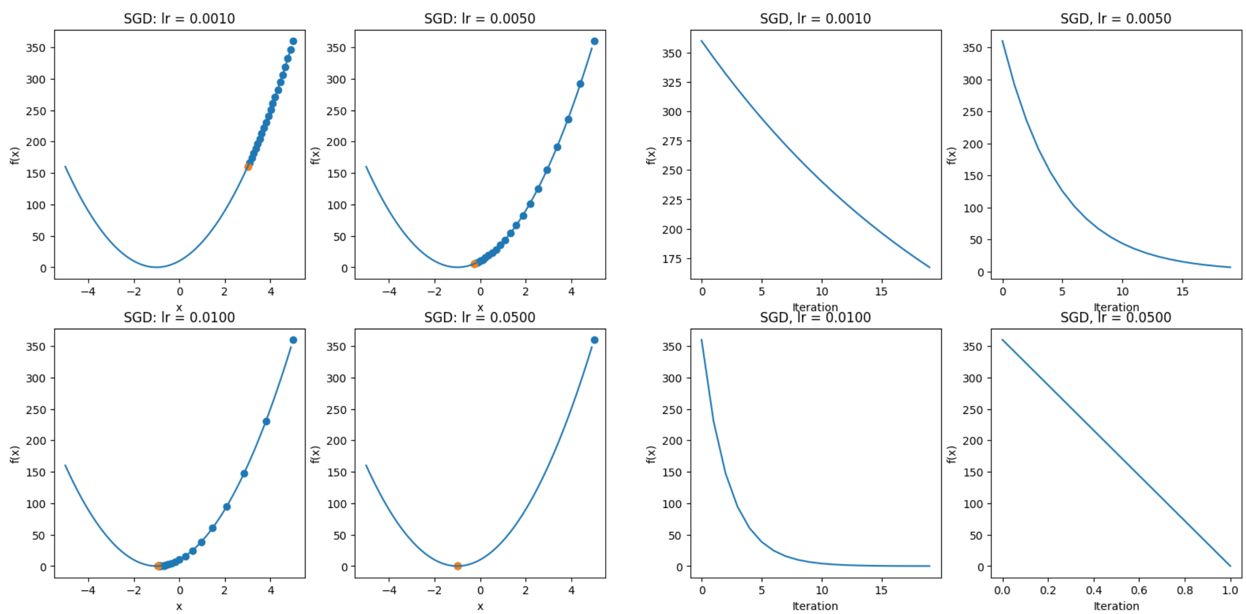
1.1 `Optimizer.sgd` method

1.1.a Test function `q1()`.

1.1.a.i Describe the termination criteria used in the `test_sgd` function in the `tests_A2.py` file. (1 mark)

Answer. There are two main termination criteria. First, after each iteration, the function checks if the magnitude of the update to the weight is smaller than the threshold (*update_thres*). If the updates have become too small, it indicates that the optimization process has likely converged to an optimal solution since the given function is a convex. Additionally, if the number of iterations reaches the maximum limit (10,000), the function terminates to prevent it from running indefinitely.

1.1.a.ii Include the figures generated by `q1()` in your `PA2_qa.pdf` file. (1 mark)



(a) SGD decision parameter trace.

(b) SGD Loss vs. iteration.

Figure 1: Figures generated by `q1()`.

1.1.a.iii With learning rate $\eta = 0.05$, what would be the value of w_1, *i.e.*, after one iteration of SGD update. Show your mathematical process. If you implemented SGD correctly, the figures generated by `q1()` should verify your w_1. (1 mark)

Answer.

$$f(w) = 10w^2 + 20w + 10$$

Compute the gradient

$$\frac{df(w)}{dw} = 20w + 20$$

Apply the SGD update rule and the way SGD updates its weight is:

$$w_{t+1} = w_t - \text{learning rate} \times \nabla f(w_t)$$

where $\nabla f(w_t)$ is the gradient computed in Step 1. We are given that $w_0 = 5$ and the learning rate is 0.05.

Compute the gradient at $w_0 = 5$:

$$\nabla f(5) = 20(5) + 20 = 120$$

Update w:

$$w_1 = w_0 - \eta \times \nabla f(w_0)$$

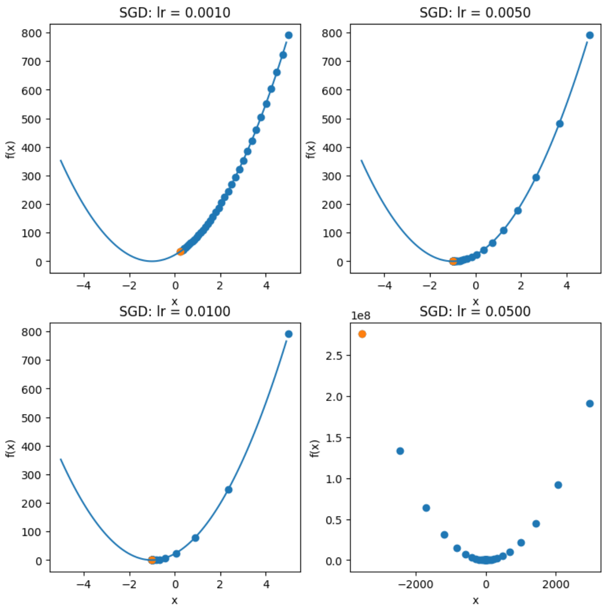
Substitute the values:

$$w_1 = 5 - 0.05 \times 120 = 5 - 6 = -1$$

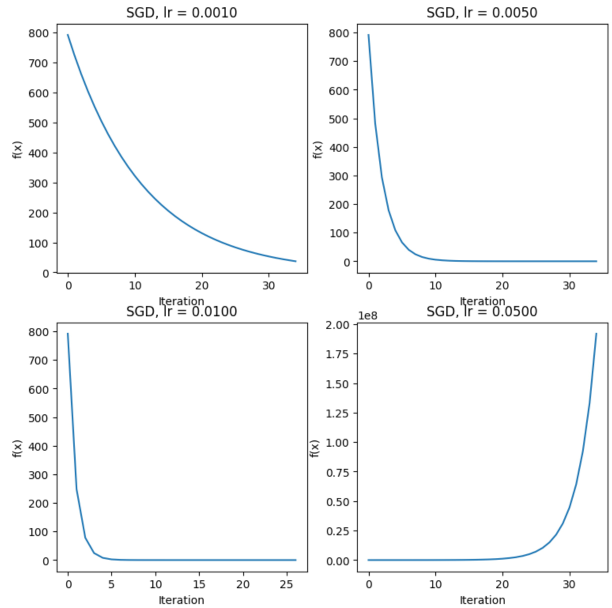
Thus, after one iteration of SGD, the new value of w is -1.

1.1.b Test function `q2()`.

1.1.b.i Include the figures generated by `q2()` in your `PA2_qa.pdf` file. (1 mark)



(a) SGD decision parameter trace.



(b) SGD Loss vs. iteration.

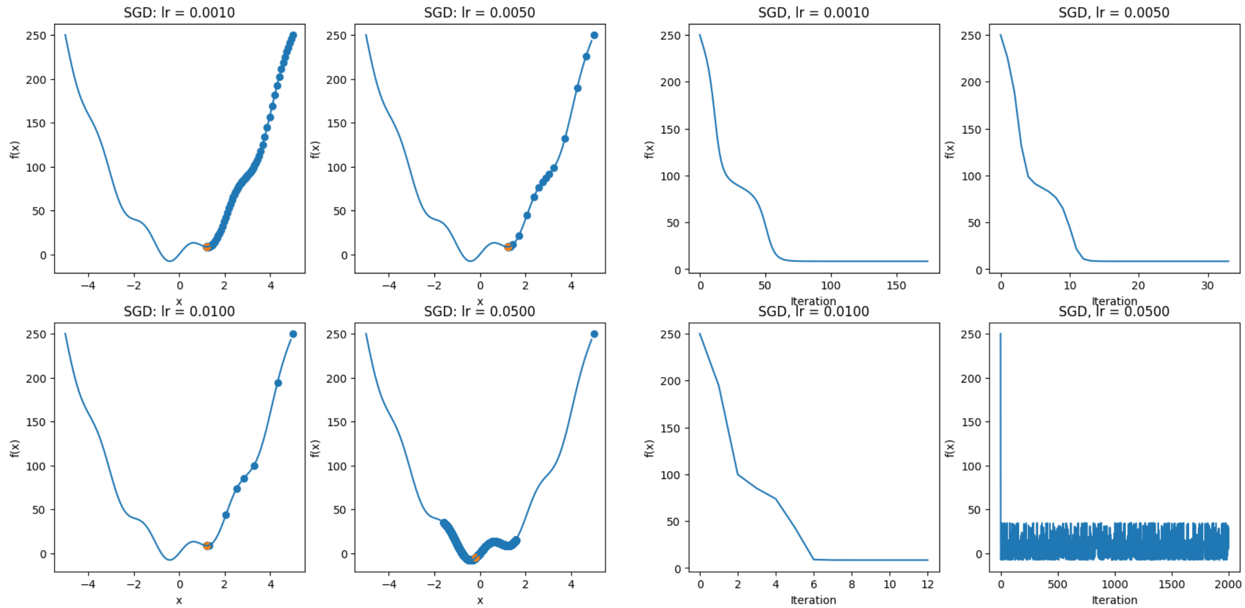
Figure 2: Figures generated by `q2()`.

1.1.b.ii When $\eta = 0.05$, SGD would fail to converge to the optimal solution. What causes such behavior? (1 mark)

Answer. When $\eta = 0.05$, the learning rate is too large. The updates overshoot the optimal solution during each iteration. Each step moves the parameter w too far from the optimum, leading to oscillation or even divergence, preventing the algorithm from converging.

1.1.c Test function `q3()`.

1.1.c.i Include the figures generated by `q3()` in your `PA2_qa.pdf` file. (1 mark)



(a) SGD decision parameter trace.

(b) SGD Loss vs. iteration.

Figure 3: Figures generated by `q3()`.

1.1.c.ii In 1-2 sentences describe the behavior of SGD in `q3()` when $\eta = 0.001, 0.005$, and 0.01. Explain why SGD fails to find the global optimum point? (1 mark)

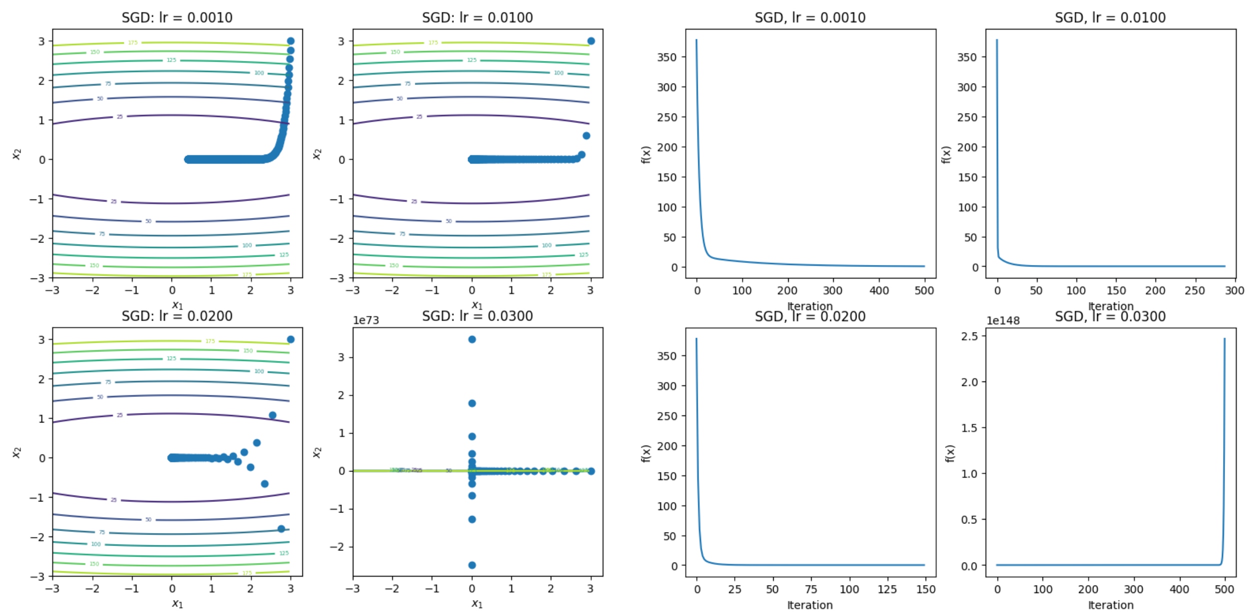
Answer. With $\eta = 0.001$, $\eta = 0.005$, and $\eta = 0.01$, those learning rates are too small and the algorithm gets stuck in local minima instead of reaching the global minimum. Since the function $f(w) = 10w^2 + 10\sin(\pi w)$ is non-convex, the gradient descent struggles to escape the local minima around the oscillations

1.1.c.iii In 1-2 sentences describe the behavior of SGD in `q3()` when $\eta = 0.05$. (1 mark)

Answer. When $\eta = 0.05$, the learning rate is too large. The updates overshoot the optimal solution during each iteration. Each step moves the parameter w too far from the optimum, leading to oscillation or even divergence, preventing the algorithm from converging.

1.1.d Test function q4().

1.1.d.i Include the figures generated by q4() in your PA2_qa.pdf file. (1 mark)



(a) SGD decision parameter trace.

(b) SGD Loss vs. iteration.

Figure 4: Figures generated by q4().

1.1.d.ii In 1-2 sentences describe the behavior of SGD in q3() when $\eta = 0.001$ and 0.01. How is this behavior related to the stretched nature of the function $f(\underline{w})$? (1 mark)

Answer. When $\eta = 0.001$ and 0.01, SGD performs slow progress along the w1 axis and faster movement along w2 axis. The gradient is much steeper in w2 direction than in w1 direction due to the stretched nature of the function, causing uneven updates and inefficient convergence.

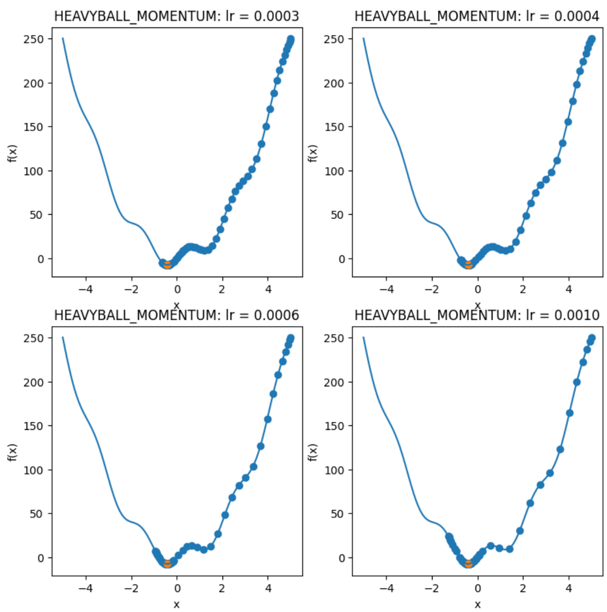
1.1.d.iii In 1-2 sentences describe the behavior of SGD in q3() when $\eta = 0.03$. (1 mark)

Answer. When $\eta = 0.03$, the learning rate is too large, so it causes oscillations, particularly in w2 direction. The steep gradient in w2 direction combined with the large step size leading to overshoot, which makes it difficult for SGD to converge to the optimal solution.

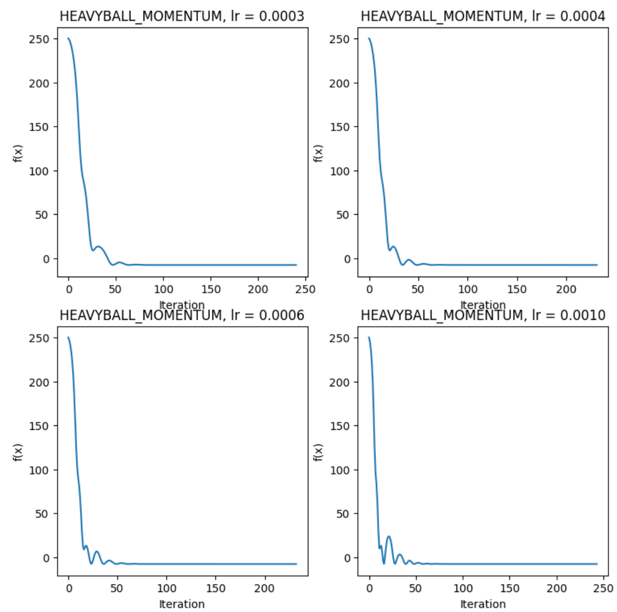
1.2 `Optimizer.heavyball_momentum` and `Optimizer.nestrov_momentum` methods

1.2.a Test function `q5()`.

1.2.a.i Include the figures generated by `q5()` in your `PA2_qa.pdf` file. (1 mark) use proper address to your png files



(a) Heavy-ball momentum decision parameter trace.



(b) Heavy-ball momentum Loss vs. iteration.

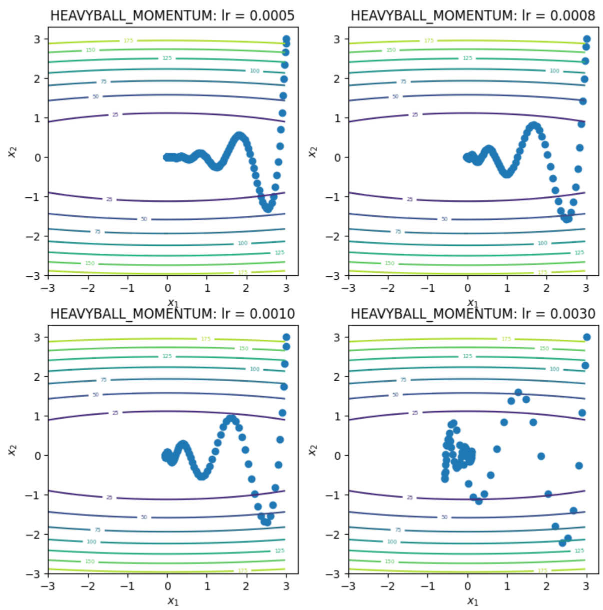
Figure 5: Figures generated by q5().

1.2.a.ii In 1-2 sentences, compare the performance of SGD with and without heavy-ball momentum by comparing the outcome of tests `q3()` and `q5()` (2 marks)

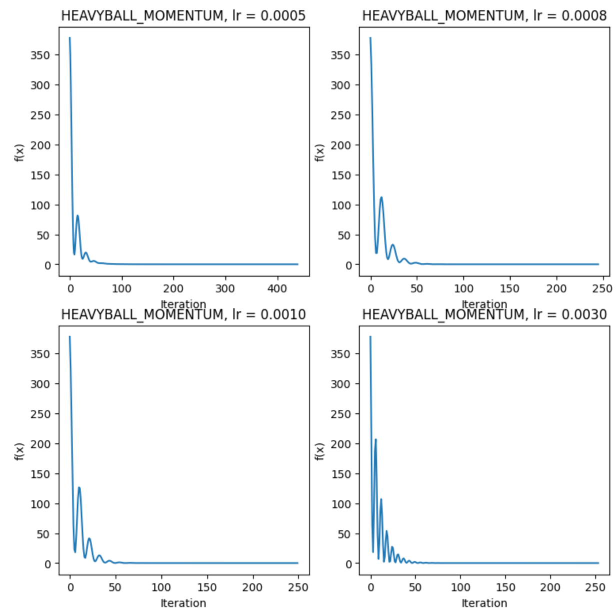
Answer. In `q3()`, SGD fails to converge due to either a small learning rate getting stuck at local minima or a large learning rate causing oscillations. In contrast, `q5()` with heavy-ball momentum enables faster convergence, as the momentum term helps overcome local minima and dampens oscillations, leading to successful convergence in all cases.

1.2.b Test function `q6()`.

1.2.b.i Include the figures generated by `q4()` in your `PA2_qa.pdf` file. (1 mark)



(a) Heavy-ball momentum decision parameter trace.

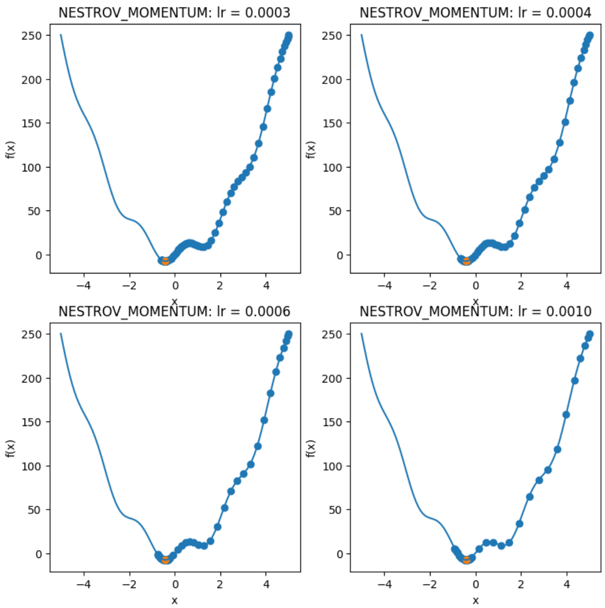


(b) Heavy-ball momentum Loss vs. iteration.

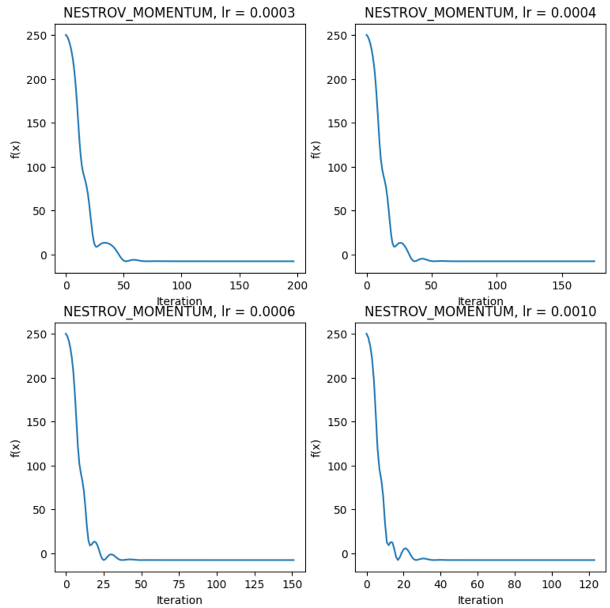
Figure 6: Figures generated by q6().

1.2.c Test function `q7()`.

1.2.c.i Include the figures generated by `q5()` in your `PA2_qa.pdf` file. (1 mark)



(a) Nestrov momentum decision parameter trace.

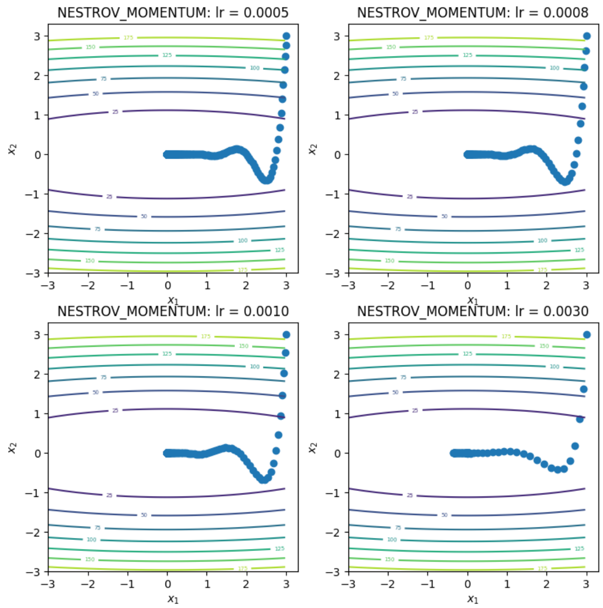


(b) Nestrov momentum Loss vs. iteration.

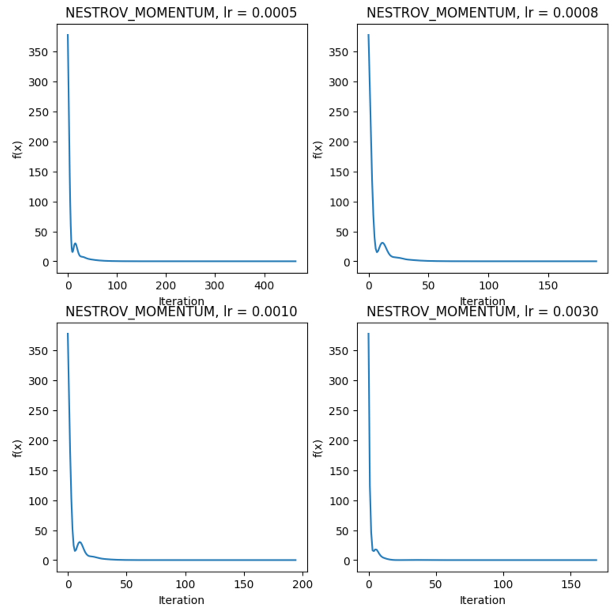
Figure 7: Figures generated by q7().

1.2.d Test function `q8()`.

1.2.d.i Include the figures generated by `q4()` in your `PA2_qa.pdf` file. (1 mark)



(a) Nestrov momentum decision parameter trace.



(b) Nestrov momentum Loss vs. iteration.

Figure 8: Figures generated by q8().

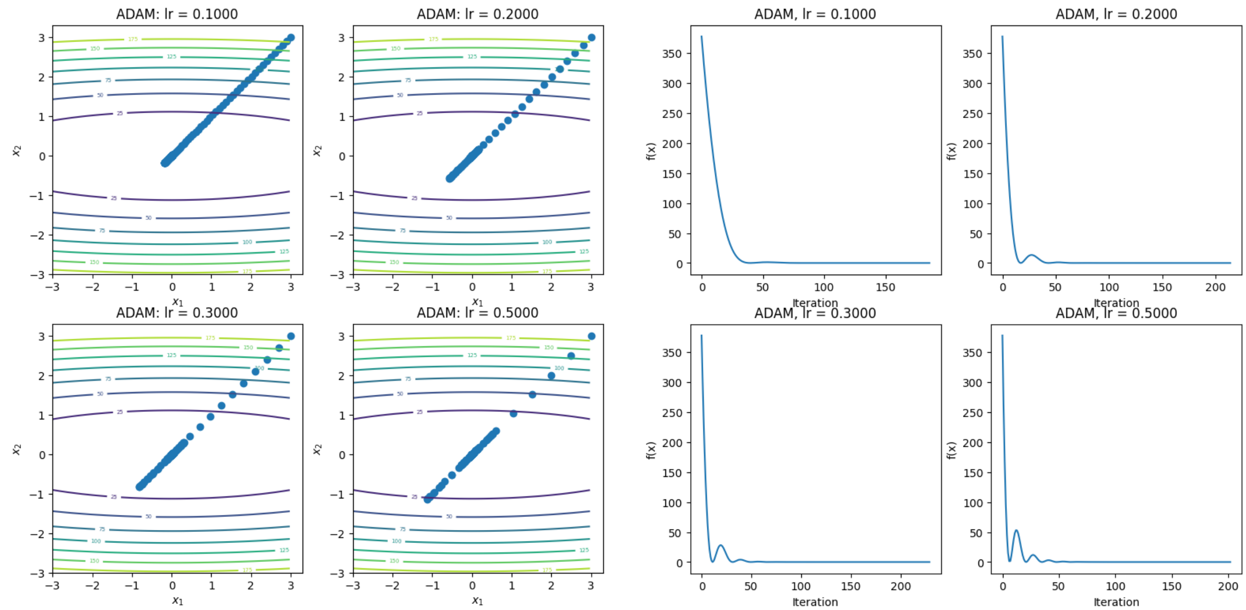
1.2.d.ii In 1-2 sentences, compare the performance of Nestrov Momentum with the heavy-ball momentum by comparing the outcome of tests `q5()` and `q6()` with that of `q7()` and `q8()`. (1 mark)

Answer. Nesterov Momentum provides faster convergence (fewer training iterations) and smoother updates in w, outperforming heavy-ball momentum in both tests. Its "lookahead" feature helps prevent overshooting and oscillations, resulting in more stable convergence compared to heavy-ball momentum.

1.3 `Optimizer.adam` method

1.3.a Test function `q9()`

1.3.a.i Include the figures generated by `q9()` in your `PA2_qa.pdf` file. (1 mark)



(a) Adam momentum decision parameter trace.

(b) Adam momentum Loss vs. iteration.

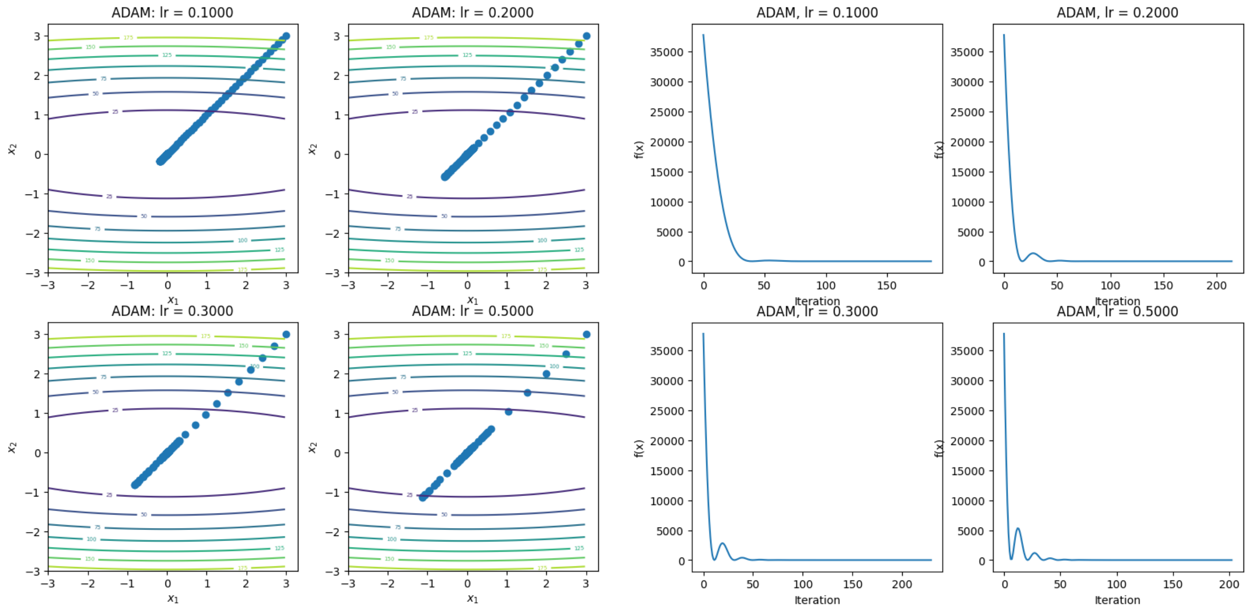
Figure 9: Figures generated by q9().

1.3.a.ii In 1-2 sentences, compare the performance of adam with momentum method (heavy-ball or Nestrov) (2 marks)

Answer. The Adam optimizer shows a faster convergence towards the optimal solution, as it shows a more rapid decrease in loss values and fewer oscillations near the minimum. Bur for momentum method, it shows a slower convergence and may oscillate more around the optimal point due to their dependence on past gradients.

1.3.b Test function `q10()`.

1.3.b.i Include the figures generated by `q10()` in your `PA2_qa.pdf` file. (1 mark)



(a) Adam momentum decision parameter trace.

(b) Adam momentum Loss vs. iteration.

Figure 10: Figures generated by q10().

1.3.b.ii Based on the outcome of `q9()` and `q10()`, describe the advantage of Adam in 1-2 sentence. (2 marks) [**HINT:** run `q11()` to see what could be the impact of scaling the function (or gradients) on the other optimization method such as gradient descent with Nestrov Momentum. You don't need to report the output of `q11()` in your report. Also, note that `q11()` would most often result in error. Don't worry. That is intentional. Try to understand why this happens.]

Answer. The advantage of Adam is its ability to adapt the learning rate for each parameter during training, which helps it maintain stable and less sensitive to scaling issues compared to Nesterov Momentum. As seen in the results of q9() and q10(), Adam consistently reduces the loss and converges smoothly, regardless of the learning rate or the function's scaling.

2 Multiclass Logistic Regression

2.1 Implementing the Learning Model

No written part.

2.2 Implementing the Learning Algorithm

2.2.a The test function `q22()` runs your implementation on the Iris dataset.

2.2.a.i Include the figures generated by `q22()` in your `PA2_qa.pdf` file. (2 marks)

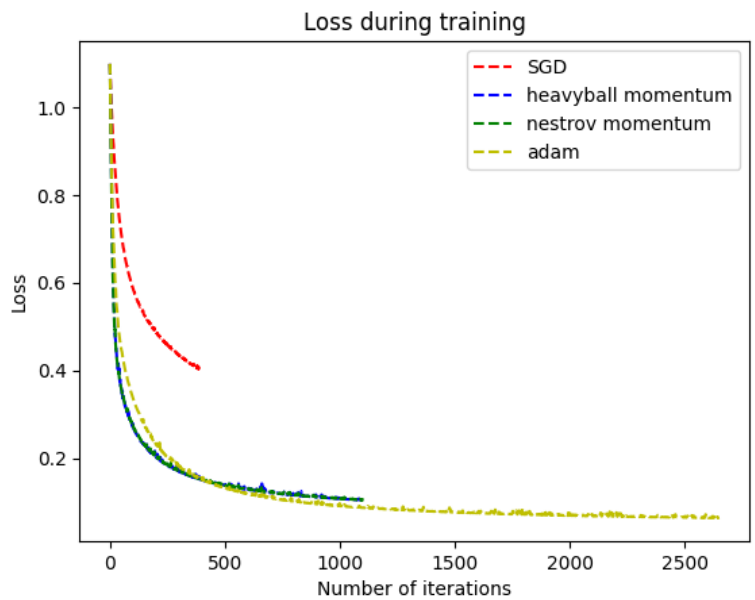


Figure 11: Figures generated by q22().

2.2.a.ii In 1-2 sentences, compare the performance of the four variants of gradient descent on this dataset (2 marks)

Answer. Adam, Nestrov Momentum, and heavyball momentum achieve lower loss values more quickly, while SGD converges more slowly and takes more iterations to reduce the loss. Overall, Adam shows the fastest and smoothest convergence.

2.2.a.iii In 1-2 sentences, explain how is it possible that the loss derived by the Adam optimizer is smaller than that of Heavy-ball Momentum, but the evaluation score of Adam is equal to the evaluation score of the heavy-ball momentum. (2 marks)

Answer. Adam optimizer achieves a smaller loss because it adapts the learning rate for each parameter, leading to more efficient updates. However, both Adam and Heavy-ball Momentum can converge to similar local/global optima in terms of classification accuracy, which leads them to be evaluated to same score.

2.2.b The test function `q23()` runs your implementation on the digits dataset.

2.2.b.i Include the figures generated by `q23()` in your `PA2_qa.pdf` file. (2 marks)

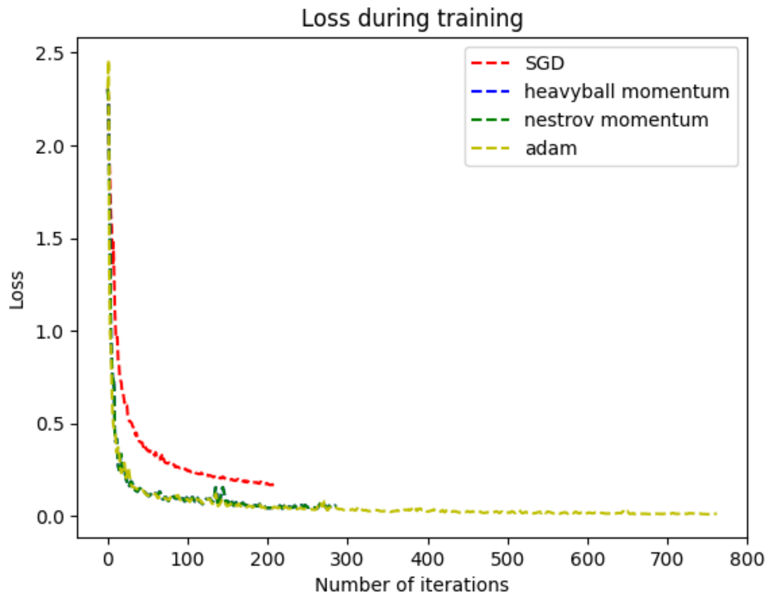


Figure 12: Figures generated by q23().

3 K-Means Clustering (Bonus)

No Written part.

4 Discussion

4.a How much time did you spend on each part of this assignment? (1 mark)

Answer. We both spend around 7-8 hours on this assignment.

4.b Any additional feedback? (optional)

Answer. No.